Graph Based Pattern Recognition

Exercise 6

Basis

- Chapter 10

Submission

- The submission takes place online on ILIAS.
- Solutions to the theory tasks must be submitted as `*.pdf` file. Other formats will not be accepted.
- Source code for the implementation tasks must be submitted as `*.py` files. Source code that we cannot compile will not be accepted.
- Individual submissions or submissions in teams of two are allowed (hand in only one copy per group. In the source code file, include the *names and matriculation numbers* of both group members in the first two lines as comments).

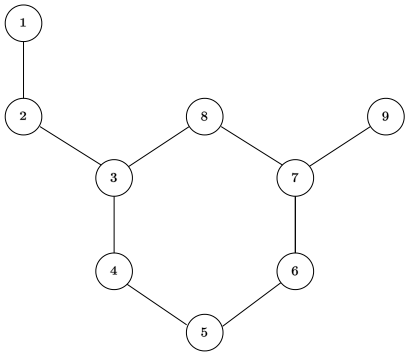
Dates

- Briefing: 10.05.2023
- Submission: 17.05.2023
- Debriefing: 17.05.2023

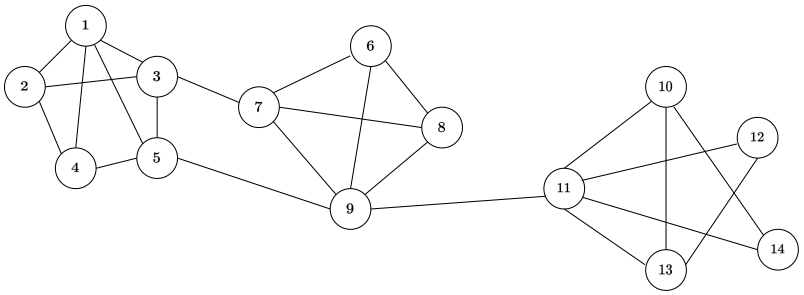
Theoretical Tasks

1. Compute the following topological descriptors for the following unlabeled graph (the numbers on the nodes reflect an arbitrary node order index):

   - *First Zagreb index*,
   - *Narumi simple topological index*,
   - *Polarity number*,
   - *Wiener index*,
   - *Randic index*, and
   - *Balaban-J index*.



2. Compute the two unary features (i.e., the *leading eigenvalues* and the *eigen-mode volumes*) as well as the *Inter-mode adjacency matrix* of the following unlabeled graph (the numbers on the nodes reflect an arbitrary node order index):



3. Apply the Spanning prototype selector with $n = 3$ to a graph data set of five graphs $\mathcal{T} = \{g_1, \ldots, g_5\}$. The distances between pairs of graphs $d_{ij} = d(g_i, g_j)$ are given by

$$\mathbf{D} = \begin{pmatrix} 0 & 1 & 3 & 7 & 11 \\ 1 & 0 & 2 & 9 & 8 \\ 3 & 2 & 0 & 6 & 14 \\ 7 & 9 & 6 & 0 & 3 \\ 11 & 8 & 14 & 3 & 0 \end{pmatrix}$$

4. Using the Prototype set P you found in the previous exercise, embed all graphs $g \in \mathcal{T}$ in $\mathbb{R}^3$. Then compute the pairwise Euclidean distance in the embedding space and compare these distances to their actual distance in the original graph domain (using the distance matrix).

Implementation Tasks

1. In this exercise series, the goal is to use a dissimilarity measure to embed graphs in a vector space. First, go to the ILIAS's webpage of the course and download/unzip `Exercise_6.zip` in your `PR_Lecture` folder. Then, navigate to `PR_lecture/Exercise_6/ex6.py` and complete the missing part of the source code.

   Create an $N \times N$ dissimilarity matrix $\mathbf{D} = (d_{ij})$ by applying a graph matching algorithm (viz. graph edit distance) to all pairs of graphs located in `PR_lecture/Exercise_6/graphs`. Then, use this dissimilarity matrix $\mathbf{D}$ to embed the N graphs into a 2D plane through MDS and generate a scatter plot to visualize them. Save the plot you obtained in `PR_lecture/Exercise_6/results/plot_mds.png`.

   Remark:

   In the file `PR_lecture/Exercise_6/graphs/classes.csv`, there is a class assigned to each graph. The first column of the class file indicates the index of the graph, while the second column indicates the corresponding class. To display the graphs, each point on the MDS plot must be colored according to its corresponding class.

   Hint:

   To compute the graph dissimilarity measure, you have the option to use either the graph edit distance algorithm you implemented in Series 2 or the algorithm provided in `networkx.algorithms.similarity`.

   The costs associated with node and edge deletion, insertion, and substitution are defined as follows:

   - $c(u \to u') = |\mu_1(u) - \mu_2(u')|$
   - $c(u \to \epsilon) = c(\epsilon \to u') = c((u, v) \to \epsilon) = c(\epsilon \to (u', v')) = \tau = 1$
   - $c((u, v) \to (u', v')) = 0$

Submission

For the coding part, you must submit a `.zip` file containing the following files. Additionally, include your solution for the theoretical tasks as `*.pdf` in the same `.zip` file.